

PCT/JP2004/008409

日 本 国 特 許 庁
JAPAN PATENT OFFICE

09.06.2004

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: 2003年 6月10日

出 願 番 号
Application Number: 特願2003-165237
[ST. 10/C]: [JP2003-165237]

出 願 人
Applicant(s): ソニー・エリクソン・モバイルコミュニケーションズ株式会
社

REC'D 29 JUL 2004

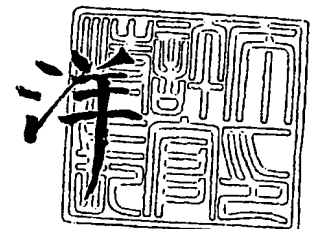
WIPO PCT

PRIORITY
DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

2004年 7月14日

特許庁長官
Commissioner,
Japan Patent Office

小 川



出証番号 出証特2004-3061010

【書類名】 特許願

【整理番号】 0300036204

【あて先】 特許庁長官殿

【国際特許分類】 G06F 9/46

【発明者】

【住所又は居所】 東京都港区港南 1 丁目 8 番 1 5 号 ソニー・エリクソン
・モバイルコミュニケーションズ株式会社内

【氏名】 大出 直樹

【特許出願人】

【識別番号】 501431073

【氏名又は名称】 ソニー・エリクソン・モバイルコミュニケーションズ株
式会社

【代理人】

【識別番号】 100107238

【弁理士】

【氏名又は名称】 米山 尚志

【手数料の表示】

【予納台帳番号】 111236

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 リソース管理方法及び装置、リソース管理プログラム、記憶媒体

【特許請求の範囲】

【請求項 1】 処理手段から送られてくるリソースの取得要求を、受付手段が受け付けるステップと、

上記処理手段からの取得要求に対応するリソースと、他の処理手段で使用するリソースとが競合するかを、競合判定手段が判定するステップと、

上記競合判定手段にてリソースが競合すると判定された各処理手段のうち、何れの処理手段に対してリソースの使用許可を与えるかを、決定手段が決定するステップと、

上記リソースの使用許可が与えられなかった処理手段から、そのリソースの開放通知要求を受け取ったとき、リスト登録手段が、当該開放通知要求に対応するリソースと当該処理手段とを対応付けてリストに登録するステップと、

上記リソースの使用許可が与えられた処理手段から当該リソースの開放完了通知を受け取ったとき、リスト確認手段が、当該開放されたリソースに対応した開放通知要求を行っている処理手段を上記リストから確認するステップと、

上記リスト確認手段が上記リストから確認した処理手段に対して、当該リソースの取得が完了したことを通知するための取得完了通知を、通知発行手段が発行するステップとを有する

ことを特徴とするリソース管理方法。

【請求項 2】 請求項 1 記載のリソース管理方法であって、

上記決定手段が上記リソースの使用許可を決定するステップでは、上記処理手段毎に予め設定されている優先度に基づいて上記リソースの使用許可を与えるか否かの決定を行うことを特徴とするリソース管理方法。

【請求項 3】 請求項 1 記載のリソース管理方法であって、

上記受付手段が上記リソースの取得要求を受け付けるステップでは、所望の処理を実行するのに必要とされる一以上のリソースをまとめた所定機能単位で、上記リソースの取得要求を受け付け、

上記競合判定手段がリソースの競合を判定するステップでは、上記所定機能単位毎にリソースの競合を判定し、

上記決定手段が上記リソースの使用許可を決定するステップでは、上記所定機能単位毎にリソースの使用許可を与えるかの決定を行い、

上記リスト登録手段が上記リストの登録を行うステップでは、上記所定機能単位毎に上記リストへの登録を行い、

上記リスト確認手段が上記リストの確認を行うステップでは、上記所定機能単位毎に上記リストの確認を行い、

上記通知発行手段が上記取得完了通知を発行するステップでは、上記所定機能単位毎に上記取得完了通知を発行することを特徴とするリソース管理方法。

【請求項 4】 請求項 3 記載のリソース管理方法であって、

上記決定手段が上記リソースの使用許可を決定するステップでは、上記所定機能単位毎に予め設定されている優先度に基づいて上記リソースの使用許可を与えるかの決定を行うことを特徴とするリソース管理方法。

【請求項 5】 処理手段から送られてくるリソースの取得要求を受け付ける受付手段と、

上記処理手段からの取得要求に対応するリソースと、他の処理手段で使用するリソースとが競合するかを判定する競合判定手段と、

上記競合判定手段にてリソースが競合すると判定された各処理手段のうち、何れの処理手段に対してリソースの使用許可を与えるかを決定する決定手段と、

上記リソースの使用許可が与えられなかった処理手段から、そのリソースの開放通知要求を受け取ったとき、当該開放通知要求に対応するリソースと当該処理手段とを対応付けてリストに登録するリスト登録手段と、

上記リソースの使用許可が与えられた処理手段から当該リソースの開放完了通知を受け取ったとき、当該開放されたリソースに対応した開放通知要求を行って、いる処理手段を上記リストから確認するリスト確認手段と、

上記リスト確認手段が上記リストから確認した処理手段に対して、当該リソースの取得が完了したことを通知するための取得完了通知を発行する通知発行手段とを有する

ことを特徴とするリソース管理装置。

【請求項 6】 請求項 5 記載のリソース管理装置であって、
上記決定手段は、上記処理手段毎に予め設定されている優先度に基づいて上記リソースの使用許可を与えるか否かの決定を行うことを特徴とするリソース管理装置。

【請求項 7】 請求項 5 記載のリソース管理装置であって、
上記受付手段は、所望の処理を実行するのに必要とされる一以上のリソースをまとめた所定機能単位で、上記リソースの取得要求を受け付け、
上記競合判定手段は、上記所定機能単位毎にリソースの競合を判定し、
上記決定手段は、上記所定機能単位毎にリソースの使用許可を与えるか決定し、
上記リスト登録手段は、上記所定機能単位毎に上記リストへの登録を行い、
上記リスト確認手段は、上記所定機能単位毎に上記リストの確認を行い、
上記通知発行手段は、上記所定機能単位毎に上記取得完了通知を発行することを特徴とするリソース管理装置。

【請求項 8】 請求項 7 記載のリソース管理装置であって、
上記決定手段は、上記所定機能単位毎に予め設定されている優先度に基づいて上記リソースの使用許可を与えるかの決定を行うことを特徴とするリソース管理装置。

【請求項 9】 処理手段から送られてくるリソースの取得要求を受け付ける受付手段と、
上記処理手段からの取得要求に対応するリソースと、他の処理手段で使用するリソースとが競合するかを判定する競合判定手段と、
上記競合判定手段にてリソースが競合すると判定された各処理手段のうち、何れの処理手段に対してリソースの使用許可を与えるかを決定する決定手段と、
上記リソースの使用許可が与えられなかった処理手段から、そのリソースの開放通知要求を受け取ったとき、当該開放通知要求に対応するリソースと当該処理手段とを対応付けてリストに登録するリスト登録手段と、
上記リソースの使用許可が与えられた処理手段から当該リソースの開放完了通

知を受け取ったとき、当該開放されたリソースに対応した開放通知要求を行って
いる処理手段を上記リストから確認するリスト確認手段と、

上記リスト確認手段が上記リストから確認した処理手段に対して、当該リソー
スの取得が完了したことを通知するための取得完了通知を発行する通知発行手段
として

コンピュータを機能させることを特徴とするリソース管理プログラム。

【請求項 10】 処理手段から送られてくるリソースの取得要求を受け付け
る受付手段と、

上記処理手段からの取得要求に対応するリソースと、他の処理手段で使用する
リソースとが競合するかを判定する競合判定手段と、

上記競合判定手段にてリソースが競合すると判定された各処理手段のうち、何
れの処理手段に対してリソースの使用許可を与えるかを決定する決定手段と、

上記リソースの使用許可が与えられなかった処理手段から、そのリソースの開
放通知要求を受け取ったとき、当該開放通知要求に対応するリソースと当該処理
手段とを対応付けてリストに登録するリスト登録手段と、

上記リソースの使用許可が与えられた処理手段から当該リソースの開放完了通
知を受け取ったとき、当該開放されたリソースに対応した開放通知要求を行って
いる処理手段を上記リストから確認するリスト確認手段と、

上記リスト確認手段が上記リストから確認した処理手段に対して、当該リソー
スの取得が完了したことを通知するための取得完了通知を発行する通知発行手段
として

コンピュータを機能させるリソース管理プログラムが記憶されたコンピュータ
読み取り可能な記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、電子機器のハードウェア資源の割り当てを管理するリソース管理方
法及び装置、リソース管理プログラム、記憶媒体に関する。

【0002】

【従来の技術】

従来より、複数の実行タスクによるハードウェア資源（以下、単にリソースとする）の利用要求が競合している場合、当該リソース要求の競合を回避するための方式としては、例えば、予め決められた優先度のみでリソースの使用権を制限する方式（以下、第1の方式とする）、一度に動作可能なアプリケーションを制限する、いわゆるシングルタスク方式（以下、第2の方式とする）、一度にリソースを利用できるのは先にリソースを取得した一つのアプリケーションだけと割り切る先取り方式（以下、第3の方式とする）、ハードウェアを抽象化した階層（レイヤー）をソフトウェアにより実現することで仮想的にリソースの同時アクセスを可能にする方式（以下、第4の方式とする）などが考えられている。

【0003】

なお、ハードウェア資源の割り付けを行う例として、特許文献1に記載された技術が知られている。この特許文献1には、マルチタスク処理において、資源への処理依頼の輻輳度に応じて待ち時間を調整することで資源の優先割り付けを行うこと、つまり、優先度の低い処理依頼を輻輳度に応じて可変的に遅らせるようにすることで優先度の高い処理依頼に対して資源を優先的に割り付ける資源割り付け方式についての技術が記載されている。

【0004】

【特許文献1】

特開平9-16416号公報（第1図）

【0005】

【発明が解決しようとする課題】

しかしながら、上述した第1の方式と第2の方式は、共に、アプリケーションの動作制限が発生してしまうため好ましくなく、そのためより動作制限の少ない方式が望まれる。第3の方式は、リソース要求の優先度がシステム内で均一でなければならず、例えば優先度に差が設けられていてもその優先度の差は無視されてしまう。また、第3の方式において、例えば優先度に応じたリソース割り付けを行う場合には、ユーザの責任による優先度の管理が必要になってしまう。第4の方式は、パーソナルコンピュータのようなハイパフォーマンス環境が必須となっ



てしまい、例えば携帯電話端末のような限られた機能と性能しか有さないものにとっては処理負担が大きすぎる。

【0006】

本発明は、このような実情に鑑みて提案されたものであり、複数アプリケーション間でハードウェア資源を効率的に且つ柔軟に割り付け可能とし、矛盾のなくリソースの排他使用を実現できる、リソース管理方法及び装置、リソース管理プログラム、記憶媒体を提供することを目的とする。

【0007】

【課題を解決するための手段】

本発明は、処理手段から送られてくるリソースの取得要求を受け付け、処理手段からの取得要求に対応するリソースと他の処理手段で使用するリソースとが競合するかを判定し、競合すると判定された各処理手段のうち何れの処理手段に対してリソースの使用許可を与えるかを決定し、リソースの使用許可が与えられなかった処理手段から、そのリソースの開放通知要求を受け取ったとき、その開放通知要求に対応するリソースと当該処理手段とを対応付けてリストに登録し、リソースの使用許可が与えられた処理手段から当該リソースの開放完了通知を受け取ったとき、その開放されたリソースに対応した開放通知要求を行っている処理手段をリストから確認し、リストから確認された処理手段に対して、当該リソースの取得が完了したことを通知するための取得完了通知を発行する。

【0008】

また、本発明によれば、リソースの取得要求の受け付け、リソースの競合判定、リソースの使用許可の決定、リストへの登録、リストの確認、取得完了通知の発行は、それぞれ、所望の処理を実行するのに必要とされる一以上のリソースをまとめた所定機能単位毎に行う。

【0009】

すなわち本発明によれば、リソースを最終的に取得するのは処理手段であるが、リソースの競合判定や使用許可、リソースの使用権利についての管理をリソース管理装置が行う。また、リソースを取得できなかった処理手段は、そのリソースが開放されたときにその旨を知らせてもらうための開放通知を要求することが



できる。リソース管理装置は、リソースを使用していた処理手段がそのリソースを開放したとき、そのリソースについて開放通知要求を行っている処理手段が存在する場合には、その処理手段に対して、リソースの開放がなされたことを通知するために取得完了通知を発行する。

【0010】

【発明の実施の形態】

以下、本発明の好ましい実施の形態について、図面を参照しながら説明する。

【0011】

本発明の具体的な構成を説明する前に、本発明にかかるリソース管理の概要を説明する。

【0012】

本発明において、ハードウェアリソース（資源、リソース）の管理は、ファンクションと呼ぶ単位で行われる。ファンクションは、本発明にかかる処理手段に相当するアプリケーションが所望する処理を実現するために使用される、一以上のリソース（以下、使用リソースとする）をまとめた機能単位であり、リソース管理の最小の管理単位である。具体例を挙げて説明すると、ファンクションは、例えば圧縮動画像の再生機能や、デジタルカメラ撮影機能等のように、電子機器の各機能毎に設定されているものである。例えば、上記圧縮動画像の再生機能のファンクションの場合は、当該圧縮動画像を再生をする上で必要な各リソースである、圧縮動画データを例えばメモリから読み出したり、通信回線を使用して取得するためのデータ取得用リソースや、その圧縮動画データを伸張する回路等の伸張処理用リソース、当該伸張処理後の再生動画像を表示するディスプレイ等の表示用リソース、同じく伸張処理後の再生音声出力するスピーカ等の音声出力用リソースなどが、使用リソースとして設定されている。また、デジタルカメラ撮影機能のファンクションの場合は、当該デジタルカメラ撮影を行う上で必要なリソースである、デジタルカメラ部等の撮影用リソースや、撮影データを圧縮等するための回路等からなる圧縮処理用リソース、圧縮後の撮影データをメモリ等に記録するための回路等からなる記録用リソースなどが、使用リソースとして設定されている。なお、このファンクションのデータは、各使用リソースを例えば

32ビットのビットマップにより表す情報となされている。

【0013】

図1には、各ファンクションと各リソースの対応関係テーブルを示す。なお、図1は各ファンクションとリソースを一般化して表しており、図1の縦列（縦軸）のFA～FNはそれぞれ異なるファンクションを示し、横列（横軸）のRA～ROはそれぞれ異なるリソースを表し、図中の丸印は縦列のファンクション内に当該丸印の付けられた横列のリソースが含まれていることを表している。すなわち図1の例の場合、ファンクションFA～FDは使用リソースとしてそれぞれリソースRCが設定されており、ファンクションFEとFFは使用リソースとしてそれぞれリソースRBが設定され、以下同様に、ファンクションFGはリソースRHが、ファンクションFHはリソースRIが、ファンクションFIはリソースRE, RG, RI, RJ, RLが、ファンクションFJはリソースRA, RC, RD, REが、ファンクションFKはリソースRF, RG, RI～RL, RNが、それぞれ使用リソースとして設定されている。ファンクションFL～FNの説明は省略する。

【0014】

図2には、本発明にかかるリソース管理を実施する電子機器の概略構成を示す。本発明実施形態の電子機器は、本発明のリソース管理装置の機能を実現するリソースマネージャ1と、電子機器が実現可能な様々な処理を実行するためのソフトウェアであるアプリケーション4と、当該電子機器が備える様々なハードウェアであるリソース（ハードウェアリソース）6と、アプリケーション4からリソース6にアクセスする際のインターフェースであるリソースアクセスライブラリ5とを有している。

【0015】

リソースマネージャ1は、本発明にかかるリソース管理を行う主要部であって、各アプリケーション4からアクセスが可能なリソース6の排他制御を行う部分であり、リソースマネージャプロセス部2と、リソースマネージャアクセスライブラリ3とを有する。リソースマネージャアクセスライブラリ3は、本発明にかかる受付手段や通知発行手段としての機能を有し、アプリケーション4からリソ

ースマネージャプロセス部2にアクセスするためのインターフェースである。したがって、アプリケーション4は当該リソースマネージャアクセスライブラリ3を使い、リソースマネージャプロセス部2に対して例えばリソースの使用権利をリクエストしたり、リソースの開放を通知したりする。リソースマネージャプロセス部2は、本発明にかかる競合判定手段、決定手段、リスト登録手段、リスト確認手段としての各機能を有しており、ハンドル管理部8とリソース管理部7を備え、各ファンクション単位でのリソース管理を行う。ハンドル管理部8は、リソースハンドルとファンクションの対応を管理し、アプリケーション4から要求されたリソースに番号を割り当て、また、アプリケーション4がどのリソースを使用しているのかを管理するため、及び競合判定の際に使用すること等のためのリストを有する。リソース管理部7は、ファンクション単位でリソースの管理を行い、図1に示した各ファンクションと各リソースの対応関係テーブルを有する。すなわち、リソース管理部7は、何れのリソースとどのリソース番号がリンクしていて、ファンクションの単位で何れのリソースが対応しているのか、さらに、何れのアプリケーションに対してリソースの使用権利を与えたりすること等の管理を行う。なお、リソースマネージャ1内のリソースマネージャアクセスライブラリ3、ハンドル管理部8、リソース管理部7は、それぞれが相互に連携して動作するため、これ以降は、それらを区別せずに、リソースマネージャ1としてまとめて説明する。

【0016】

この図2の構成において、上記アプリケーション4は、それぞれ一つ以上のファンクションを要求することが可能であり、当該ファンクションの要求により、そのファンクションの使用リソースの確保を、リソースマネージャ1へ要求する。そして、例えば二つ以上のアプリケーション4から、同時若しくは異なる時期に、上記ファンクションの要求による使用リソースの確保要求、つまり取得要求がなされた場合、リソースマネージャ1は、図1の対応関係テーブルにより、それらファンクション同士の使用リソースの同否を比較することによって、リソースの競合判定を行う。そして、リソースマネージャ1は、そのリソースの競合判定でリソースの競合が起きると判定した場合には、後述するリソース競合処理を



行う。一方、リソースの競合が起こらないと判定した場合、リソースマネージャ1は、各アプリケーション4にそれぞれリソースの取得許可を与える。

【0017】

ここで、リソースを取得する際、アプリケーション4は、取得したいリソースのファンクション名、優先度、リソースの取得要因、アクセスハンドル、プロセスIDの各情報を、リソースマネージャ1に対して指定する。上記優先度は、アプリケーション毎の優先度、ファンクション毎の優先度、リソース毎の優先度が、各アプリケーションにより適宜決められており、リソースマネージャ1は、それらの優先度の情報を、リソースの競合が発生した場合の競合制御に用いる。取得要因の情報は、リソースの確保が失敗した場合やリソース利用の権利の移動が発生した場合に、リソースマネージャ1が、そのリソースを使用するアプリケーション4に対して、当該リソース確保の失敗要因やリソース利用の権利移動の発生要因を知らせるために用いられる。ハンドルは、複数ファンクションを同時にアクセスするために用いられる情報である。プロセスIDは、リソースを取得するアプリケーション4の識別に用いられる情報である。なお、本実施形態において、リソースの取得が成功するのは、指定された全てのファンクションの使用リソースが取得できた場合であり、その他は全て取得失敗となる。取得が失敗した場合、既に一部のリソースの取得が成功していたとしても、そのファンクションは開放される。そして、リソースマネージャ1は、アプリケーション4に対して、それら取得結果をファンクション毎に通知する。

【0018】

一方、使用されなくなったリソースは速やかに開放する必要があり、当該リソースの開放は、そのリソースを使用していたアプリケーション4によりなされる。すなわち、取得されたリソースは、そのリソースを取得したアプリケーション4以外のものにより行われることはない。そして、リソースの開放を行ったアプリケーション4は、リソースマネージャ1に対して、リソースの開放完了を通知する。このとき、リソースの開放完了通知には、リソース取得時に用いたアクセスハンドルとプロセスIDが含まれる。なお、ファンクションは同時に複数指定することができるものであるが、処理自体はファンクション毎に行われるため、



開放完了通知はファンクション毎の通知となる。

【0019】

また、アプリケーション4は、ファンクションの使用リソースが全て開放されたときに、その旨を知らせてもらうための開放通知要求を、リソースマネージャ1に対して送ることができる。リソース開放通知要求にかかるファンクションは複数同時に指定でき、同時に指定された全てのファンクションの使用リソースが全て開放された時に、リソースマネージャ1は上記開放通知要求を行ったアプリケーション4に対して上記開放通知を発行する。上述の開放通知要求を送る場合、アプリケーション4は、開放通知を要求するファンクション名、開放待ちの優先度、通知待ちの時間、アクセスハンドル、プロセスIDの各情報を、リソースマネージャ1に対して指定する。なお、リソースマネージャ1が開放通知をアプリケーション4に送信するのは、指定された全てのファンクションの使用リソースが全て開放された時となる。また、通知待ち時間内にリソースが開放されない場合、リソースマネージャ1は、開放通知タイムアウト通知をアプリケーション4に送信する。リソースの開放待ちの解除は、その開放通知要求を行ったアプリケーション4が、リソースマネージャ1に対してリソース取得時に用いたアクセスハンドルとプロセスIDを指定することにより行う。つまり、当該リソースの開放待ちの解除は、リソース開放待ちのアプリケーション4以外からは行うことができない。

【0020】

そして、アプリケーション4は、開放通知を受け取った場合、速やかにリソースを要求するか、又は、リソースが不要であることをリソースマネージャ1に通知しなければならない。ここで、リソースが要求された場合、リソースマネージャ1は、更に次の通知先アプリケーションへのリソース開放通知については、次のリソース開放時に行う。一方、リソースが不要であるとされた場合、リソースマネージャ1は、次の通知先アプリケーションへリソース開放通知を行う。なお、アプリケーション4がそれらリソースの要不要の通知を怠った場合、リソースマネージャ1は、一定時間経過でタイムアウトとし、リソースが不要であるとみなして、次の通知先アプリケーションへリソース開放通知を行う。また、リソ



ースマネージャ1は、開放通知要求元のアプリケーションによる指定時間を過ぎても、リソースを使用しているアプリケーションから開放完了の通知が来ない場合に、開放要求の待ち状態をキャンセルし、その通知要求元のアプリケーションにタイムアウト通知を行う。また、複数の開放通知要求が存在する場合、それらの開放通知の通知順は、優先度大>優先度小、設定順後>設定順前とし、初めに優先度で選別し、同一優先度のものが複数存在するならば、設定順の規則により通知先を決定する。

【0021】

なお、リソースの取得要求及び開放要求、開放通知要求は、ファンクションを複数同時に指定することができるが、処理は各ファンクション毎に個別に実施され、その結果の通知はファンクション毎となる。このため、複数のファンクションを同時に指定した場合、その結果はファンクション毎に異なるものとなる。

【0022】

その他、リソースマネージャ1は、登録されたアプリケーションに対して、定期的にヘルスチェックを行うことができる。当該ヘルスチェックによって、アプリケーションの消滅が確認された場合、リソースマネージャ1は、そのアプリケーションの使用リソースを全て開放し、当該アプリケーションに関する登録内容を解除する。

【0023】

以下、リソースの競合制御時の具体的な動作を個々に説明する。

【0024】

アプリケーション4がファンクションを要求した際に、そのファンクションに既に別のアプリケーションにより使用されているリソースが含まれているために、当該ファンクションに対する取得要求を実施すると競合が発生してしまう場合、リソースマネージャ1は、優先度の高いアプリケーションにリソースを使用する権利を与える。優先度が同一の場合、リソースマネージャ1は、最後に取得要求を行ったアプリケーションに、リソースを利用する権利を与える。また、リソースの利用権利の移動が発生した場合は、リソースマネージャ1は、そのリソースを使用するアプリケーションにその要因を知らせ、それに対する応答を待つ。



なお、権利の移動は、要因通知に対する応答返却、又は要因通知後に開放要求が送られてきた時に行われる。

【0025】

アプリケーション4が同時に複数のファンクションを取得する場合、リソースマネージャ1は、無駄な開放処理を避けるために、その中にリソース競合により取得が失敗するものが一つでも含まれているならば、当該アプリケーション4に対して、取得失敗として応答を一つだけ通知する。このとき、アプリケーション4は、指定した全てのファンクションのリソース取得を実施しない。一方、リソース競合により取得が失敗しない場合、リソースマネージャ1は、指定されたファンクション毎に取得の判定処理を実施し、それら各ファンクション毎の取得結果をアプリケーション4に通知する。

【0026】

アプリケーション4が開放通知要求を行っている場合において、開放要求通知の応答の待ち時間内に、同一のリソース要求を受信した場合、リソースマネージャ1は、それら受け付けた要求のうち、より優先度が高く、より後に要求したアプリケーションに対してリソースを取得させる。

【0027】

また、リソース取得の権利移動のための要因通知に対する応答又は開放要求が、アプリケーション4がリソース取得要求時に指定した「開放処理時間」内に通知されない場合、リソースマネージャ1は、その取得要求を行ったアプリケーションと、リソースを使用しているアプリケーションに対して、「開放異常」を通知する。開放異常となった場合、リソースマネージャ1は、リソースが開放されない限り、その後の取得要求を全て開放異常とする。

【0028】

次に、図2の各部間で通信されるデータについて説明する。

【0029】

アプリケーション4からリソースマネージャアクセスライブラリ3へは、初期設定要求データ、終了設定要求データ、内部と外部同期用のリソース取得要求データ、リソース開放要求データ、リソース開放通知要求データ、リソース開放通



知要求取り消しデータ等が送られる。初期設定要求データは、メッセージ受信方法と受信先ポインタを示すデータからなる。リソース取得要求データは、各ファンクション名からなるファンクション群データ、優先度データ、取得要因データ、リソース開放最大処理時間データ、リソースハンドルからなる。リソース開放要求データは、リソースハンドルからなる。リソース開放通知要求データは、ファンクション群データ、優先度データ、リソースハンドルからなる。リソース開放通知要求取り消しデータはリソースハンドルからなる。ここで、優先度データは、取得要求及び開放通知要求の優先度を示すデータである。取得要因データは、リソースの取得要因を表すデータである。リソース開放最大処理時間データは、実際に使用リソースをアプリケーションが開放するのにかかる最大時間を表すデータである。

【0030】

アプリケーション4からハンドル管理部8へは、プロセスID、メッセージ種別データ、メッセージデータの各データが送られる。プロセスIDはアプリケーションのプロセスIDである。メッセージ種別データは初期設定要求メッセージ、終了設定要求メッセージ、取得要求メッセージ、開放要求メッセージ、開放通知要求メッセージ、開放通知解除要求メッセージの各種がある。メッセージデータは、メッセージ種別に応じた各メッセージの内容を示すデータである。なお、メッセージ種別において、初期設定要求メッセージはアプリケーションの登録のためのメッセージであり、終了設定要求メッセージはアプリケーションの登録を解除するためのメッセージ、取得要求メッセージはファンクションの取得を要求するためのメッセージ、開放要求メッセージはファンクションの開放を要求するためのメッセージ、開放通知要求メッセージはリソース開放を待つためのメッセージ、開放通知解除要求メッセージはリソース開放待ちを解除するためのメッセージである。また、初期設定要求メッセージの内容としてはメッセージの受信方法と受信ポインタがあり、取得要求メッセージの内容としてはリソースハンドル、取得ファンクション群、取得優先度、取得要因、リソース開放最大処理時間、リソース取得に対する最初の応答メッセージの送信先等がある。開放要求メッセージの内容としては開放するリソースハンドルがあり、開放通知要求メッセージ



の内容としては開放通知用のリソースハンドル、開放を待つファンクション群、開放通知の優先度、開放通知の待ち時間がある。開放通知解除要求メッセージの内容としては取り消す開放通知のためのリソースハンドルがある。

【0031】

ハンドル管理部 8 からアプリケーション 4 へは、リソースハンドルデータ、メッセージ種別データ、要因データが送られる。要因データは、要求の失敗及び開放要求通知の開放要求発生要因のデータがある。メッセージ種別データは、取得結果 OK の通知メッセージ、取得結果 NG 通知メッセージ、開放要求通知メッセージ、開放通知メッセージ、開放通知待ちタイムアウト通知メッセージ、開放異常通知メッセージがある。取得結果 OK 通知メッセージはリソースの取得に成功したことを示す取得要求許可のメッセージであり、取得結果 NG 通知メッセージはリソースの取得に失敗したことを示す取得 NG のメッセージである。開放要求通知メッセージはハンドルの開放実施の要求のメッセージ、開放通知メッセージは指定リソースが全て開放されたことを示すメッセージ、開放通知待ちタイムアウト通知メッセージは開放通知待ち時間を経過してもリソースが開放されなかったことを通知するメッセージである。開放異常通知メッセージは取得要求で発生した開放通知に対する開放要求がタイムアウトしたことを通知するメッセージであり、取得要求アプリケーションと開放要求アプリケーションの双方に通知される。

【0032】

ハンドル管理部 8 からリソース管理部 7 へは、メッセージ ID、メッセージ種別データ、ファンクション群データ、プロセス ID、優先度データ、要因データ、リソース開放時間データ、開放待ち時間データの各データが送られる。メッセージ ID は、メッセージの識別子である。送信メッセージに対する応答メッセージが存在する場合、応答メッセージのメッセージ ID は送信メッセージの ID となる。メッセージ種別データは、取得要求メッセージ、開放要求メッセージ、不用応答メッセージ、開放通知要求メッセージ、要因応答メッセージの各種がある。ファンクション群データは、ファンクションの群を示すデータである。プロセス ID は、アプリケーションプロセスの ID である。優先度データは、取得要求



及び開放通知要求の優先度を示すデータである。要因データは、リソースの取得要因を表すデータである。リソース開放時間データは、実際に使用リソースをアプリケーションが開放するのにかかる最大時間を表すデータである。開放通知待ち時間データは、開放通知要求から開放通知を受け取るまでのタイムアウト時間を表すデータである。なお、メッセージ種別において、取得要求メッセージはファンクションの取得を要求するためのメッセージであり、開放要求メッセージはファンクションの開放を要求するメッセージ、不用応答メッセージは開放通知受信時にそのファンクション群が不用であることの応答に用いるメッセージ、開放通知要求メッセージは指定ファンクションの全リソースが開放されたことを通知するよう依頼するメッセージ、要因応答メッセージは要因通知に対する応答のメッセージである。

【0033】

リソース管理部7からハンドル管理部8へは、メッセージID、メッセージ種別データ、ファンクション群データ、プロセスID、要因データ、要因プロセスデータ、要因優先度データの各データが送られる。メッセージIDは、メッセージの識別子であり、ハンドル管理部の送信メッセージのメッセージIDとなる。メッセージ種別データは、取得結果OK通知メッセージ、取得結果NG通知メッセージ、開放結果OK通知メッセージ、開放結果NG通知メッセージ、要因通知メッセージ、開放通知メッセージ、開放通知タイムアウト通知メッセージ、開放異常通知メッセージの各種がある。ファンクション群データは、ファンクションの群を示すデータである。プロセスIDは、アプリケーションプロセスのIDである。要因データは、リソースの取得失敗、開放失敗、要因通知発生の要因及び開放通知ファンクションの開放前の取得要因の各データである。要因プロセスデータは、リソースの取得失敗、開放失敗、要因通知発生の要因となったプロセス及び開放通知ファンクションの開放以前の取得プロセスのIDの各データである。要因優先度データは、リソースの取得失敗、開放失敗、要因通知発生の要因となった優先度及び開放通知ファンクションの開放以前の取得優先度のデータである。なお、メッセージ種別において、取得結果OK通知メッセージはファンクションの取得が成功したことを示すメッセージであり、取得結果NG通知メッセー



ジはファンクションの取得に失敗したことを示すメッセージ、開放結果OK通知メッセージはファンクションの開放が成功したことを示すメッセージ、開放結果NG通知メッセージはファンクションの開放に失敗したことを示すメッセージ、要因通知メッセージはファンクションリソースの使用権が移動することを通知するメッセージ、開放通知メッセージは開放通知要求の指定ファンクションの全リソースが開放されたことを通知するメッセージ、開放通知タイムアウト通知メッセージは開放通知要求がタイムアウトしたことの通知のメッセージ、開放異常通知メッセージは取得要求で発生した要因通知に対する要因応答がタイムアウトしたことの通知のメッセージであり、要因通知方向と取得要求方向の双方に送信される。

【0034】

図3は、本実施形態の電子機器の一例としての携帯電話端末の概略構成を示す。なお、この図3において、図2に対応する各構成要素には同じ指示符号を付している。また、図3の各構成要素は、携帯電話端末の主要な構成のみを示している。

【0035】

図3の例において、携帯電話端末のアプリケーション4は、携帯電話端末の電話機能を実現する電話アプリケーション(Tel App)43、時計機能を実現するクロックアプリケーション(Clock)44、ディスプレイ上の表示とスクリーンセーブ機能を実現するスクリーンアプリケーション(Screen App)45、アプリケーションを起動させるためのアプリケーションランチャー(App Launcher)46の他に、ユーザにより定義されたアプリケーション(APP1, APP2)41, 42等からなる。それら各アプリケーションは、アプリケーションの基本部分であるアプリケーションフレームワーク(Application Framework)14、ウインドウマネージャ(Window Manager)13を介し、さらに図2のリソースアクセスライブラリ5を介してオペレーティングシステム(OS)21に接続されている。

【0036】

図3の例の場合、リソース6は、例えば表示を行うための液晶ディスプレイ(LCD)デバイス62とそのドライバ61、テンキー等のキー(KEY)デバイス64

とそのドライバ63、携帯電話端末における通信のための送受信(Communication)及びその他(Other)の主要な機能に対応したシステムデバイス66とそのドライバ65、デジタルカメラのカメラ(Camera) デバイス68とそのドライバ67、本発明の記憶媒体としての機能を有するメモリ(Mem)とそのメモリへのファイル(File)書き込み読み出し等を行うメモリデバイス72とそのドライバ71と、その他、LED(発光ダイオード)やオーディオデバイス等の各種デバイス(Misc H/W)70とそのドライバ69等からなる。これら各デバイスの制御や各種の演算は、CPU(Central Processing Unit)23が行う。

【0037】

また図3において、タスクマネージャ(Task Manager)11は、アプリケーション4の実行時の各タスクを管理し、イベントマネージャ12(Event Manager)は各種イベントを管理する。リソースマネージャ(Resource Manager:ResMan)1は、図2のリソースマネージャ1に対応する。

【0038】


以下、上述した本実施形態のリソースマネージャが実際にどのようにして各リソースの管理を行うのかの一例を、図4～図8の各タイムフローチャートを用いて説明する。

【0039】

図4は、アプリケーションAPP2より先に、アプリケーションAPP1がファンクション指定によるリソース取得要求を行っているが、優先度はアプリケーションAPP1がアプリケーションAPP2よりも高く($APP1 > APP2$)、更にアプリケーションAPP2が開放通知要求を行わない場合における、本実施形態のリソースマネージャ(ResMan)が行うリソース(Resource)の競合管理処理のタイムフローチャートを示している。

【0040】

図4において、例えばアプリケーションAPP1から、ステップS1にてファンクションの指定による新規のリソース取得要求を受け取ると、リソースマネージャは、ステップS2にて競合判定を行う。このとき、そのファンクションの使用リソースは他のアプリケーションにより利用されていないため、リソースマネ



ージャは、ステップS3にて、アプリケーションAPP1に対して取得要求許可を返す。これにより、アプリケーションAPP1は、ステップS4にて所望のリソースを取得できることになる。

【0041】

その後、別のアプリケーションAPP2から、ステップS5にて、同じ使用リソースを含むファンクションの指定による新規のリソース取得要求を受け取ると、リソースマネージャは、ステップS6にて競合判定を行う。このとき、そのファンクションの使用リソースは既にアプリケーションAPP1により利用されているため、リソースマネージャは、ステップS7にて、アプリケーションAPP2に対して取得NG通知を返す。これにより、アプリケーションAPP2は、ステップS8にて所望のリソースを取得できなかったとしてNG処理を行う。

【0042】

図5は、アプリケーションAPP2より先にアプリケーションAPP1がファンクション指定によるリソースの取得要求を行っているが、優先度はアプリケーションAPP1がアプリケーションAPP2よりも高く（APP1>APP2）、更にアプリケーションAPP2が開放通知要求を行う場合における、本実施形態のリソースマネージャが行うリソースの競合管理処理のタイムフローチャートを示している。なお、図5において、図4と同じ処理のステップについてはその説明を省略する。

【0043】

図5において、ステップS8によりNG処理を行った後、アプリケーションAPP2は、開放通知要求をリソースマネージャに送る。このときのリソースマネージャは、当該開放通知要求にかかるファンクションとアプリケーションを対応つけてリストに登録する。その後、アプリケーションAPP1にて所望の処理が終了し、ステップS10にてリソースの開放が行われ、ステップS11にてアプリケーションAPP1から開放完了通知を受け取ると、リソースマネージャは、ステップS12にて、他のアプリケーションによる開放通知要求があるか否かの開放通知要求検索を行う。この例の場合、アプリケーションAPP2からの開放通知要求がリストに登録されているため、リソースマネージャは、ステップS1

3にて当該アプリケーションAPP2に対してリソースが開放されたことを通知(開放通知)する。

【0044】

これにより、アプリケーションAPP2は、リソースの開放がなされたことを知ることができる。リソースの開放がなされたことを知ったアプリケーションAPP2は、ステップS14にてファンクションの指定による再度の取得要求を行う。当該ステップS14にて取得要求を受け取ると、リソースマネージャは、ステップS15にて競合判定を行う。このとき、当該ファンクションの使用リソースは他のアプリケーションにより利用されていないことになるため、リソースマネージャは、ステップS16にて、アプリケーションAPP2に対して取得完了通知を返す。そして、アプリケーションAPP1は、ステップS17にて所望のリソースを取得できることになる。

【0045】

図6は、アプリケーションAPP2より先にアプリケーションAPP1がファンクション指定によるリソース取得要求を行っているが、優先度はアプリケーションAPP1とアプリケーションAPP2で同じであるため取得要求の設定順の遅い方が優先され($APP1 \leq APP2$)、更にアプリケーションAPP1が開放通知要求を行わない場合における、本実施形態のリソースマネージャが行うリソースの競合管理処理のタイムフローチャートを示している。なお、図6において、図4、図5と同じ処理のステップについてはその説明を省略する。

【0046】

図6の例の場合、リソースマネージャは、ステップS6での競合判定の際に、アプリケーションAPP2の優先度がアプリケーションAPP1の優先度以上であり、アプリケーションAPP1からのリソース取得要求の設定順が「後」であるため、ステップS21にて、アプリケーションAPP1に対してリソースの開放要求を発行する。アプリケーションAPP1は、当該開放要求を受け取った場合、ステップS22及びステップS23にてリソースの開放処理を行う。その後、アプリケーションAPP1は、ステップS24にて、リソースマネージャへ開放完了通知を送る。

【0047】

アプリケーションAPP1から開放完了通知を受け取ると、リソースマネージャは、ステップS25にて、アプリケーションAPP2に対して取得完了通知を送信する。これにより、アプリケーションAPP2は、ステップS26にて所望のリソースを取得できることになる。

【0048】

その後、アプリケーションAPP2にて所望の処理が終了し、ステップS27にてリソースの開放が行われ、ステップS28にてアプリケーションAPP2から開放完了通知を受け取ると、リソースマネージャは、ステップS29にて、その開放されたリソースをリストに登録する。

【0049】

図7は、アプリケーションAPP2より先にアプリケーションAPP1がファンクション指定によるリソース取得要求を行っているが、優先度はアプリケーションAPP1とアプリケーションAPP2で同じであるため取得要求の設定順の遅い方が優先され ($APP1 \leq APP2$)、更にアプリケーションAPP1が開放通知要求を行う場合における、本実施形態のリソースマネージャが行うリソースの競合管理処理のタイムフローチャートを示している。なお、図7において、図4、図5、図6と同じ処理のステップについてはその説明を省略する。

【0050】

図7の例の場合、アプリケーションAPP1は、ステップS23にてリソースの開放を行った後、ステップS30にて開放完了通知と開放通知要求をリソースマネージャへ送る。

【0051】

その後、リソースマネージャは、ステップS28にてアプリケーションAPP2から開放完了通知を受け取ると、ステップS31にて開放待ちのリストの内容を確認する。このとき、リソースマネージャは、開放待ちリストを元にアプリケーションAPP1による開放通知要求があることを知り、ステップS32にてリソース開放通知を当該アプリケーションAPP1へ送信する。

【0052】

リソースマネージャからリソース開放通知を受け取ったアプリケーションAPP1は、ステップS33にてリソースの再取得のための再登録準備を行い、当該準備が整うと、ステップS34にて取得要求をリソースマネージャへ送る。

【0053】

そして、リソースマネージャは、ステップS35にて競合判定を行う。このとき、リソース競合は無いため、リソースマネージャは、ステップS36にて、アプリケーションAPP1に対して取得完了通知を返す。これにより、アプリケーションAPP1は、ステップS37にて所望のリソースを再取得できることになる。

【0054】

図8は、アプリケーションAPP1とAPP2の他に更にアプリケーションAPP3があり、最初にアプリケーションAPP2がリソースの取得要求を行い、次にアプリケーションAPP2が、更にその次にアプリケーションAPP3がリソースの取得要求を行い、また、優先度はアプリケーションAPP3が最も高く、アプリケーションAPP1とAPP2の優先度は同じであるため取得要求の設定順の遅い方が優先され ($APP1 \leq APP2 < APP3$)、更にアプリケーションAPP1が開放通知要求を行う場合における、本実施形態のリソースマネージャが行うリソースの競合管理処理のタイムフローチャートを示している。なお、図8において、図4、図5、図6、図7と同じ処理のステップについてはその説明を省略する。

【0055】

図8の例の場合、アプリケーションAPP3の取得要求は、ステップS5でアプリケーションAPP2による取得要求の後に行われることになる。リソースマネージャは、ステップS40にてアプリケーションAPP3からのリソースの取得要求を受け取ると、アプリケーションAPP1からステップS30の開放完了通知及び開放通知要求の送信を受け取った後に、ステップS41にてアプリケーションAPP1、APP2、APP3間でのリソース競合の判定を行う。この場合、リソースマネージャは、優先度がもっとも高いのがアプリケーションAPP3であるため、ステップS42にて当該アプリケーションAPP3に取得完了通



知を送る。またこの時のリソースマネージャは、アプリケーションAPP2に対しては、ステップS43にて取得NGを送る。これにより、アプリケーションAPP3は、ステップS44にて所望のリソースを取得できることになる。

【0056】

その後、アプリケーションAPP3は、所望の処理が終了すると、ステップS45にてリソースの開放を行い、更にステップS46にて開放完了通知をリソースマネージャへ送る。

【0057】

開放完了通知を受け取ったリソースマネージャは、ステップS47にて開放待ちのリストの内容を確認する。このとき、リソースマネージャは、開放待ちリストから、アプリケーションAPP1が開放通知要求を行っていることを知る。その後は、図7のステップS32～ステップS37と同様の処理が行われる。

【0058】

以上説明したように、本発明実施形態のリソースマネージャは、使用状況（条件）に応じた動的な優先度管理と使用権管理を実現することで、柔軟なリソース管理機構を実現している。すなわち、本発明実施形態のリソースマネージャは、リソースの使用権を管理することで、実際のハードウェアリソースの利用アプリケーションと使用権利アプリケーションを柔軟に定義できるようにしている。

【0059】

また、本実施形態によれば、使用権管理を行う単位として、個別のハードウェアリソースの単体から、或るアプリケーションを構成するために必要な複数リソースをまとめた単位としてファンクションという概念を導入し、このファンクション単位での管理を可能としている。

【0060】

また、本実施形態によれば、使用アプリケーションの優先度、リソース優先度、などを考慮したリクエストキューを内部に持つことで、単純な後優先アルゴリズムよりも無駄なリソースの奪い合いの回数を減らすことも出来るようになってくる。

【0061】

なお、上述した実施形態の説明は、本発明の一例である。このため、本発明は上述した実施の形態に限定されることなく、本発明に係る技術的思想を逸脱しない範囲であれば、設計等に応じて種々の変更が可能であることはもちろんである。本発明は、携帯電話端末だけでなく、例えばパーソナルコンピュータやPDA装置（PDA: Personal Digital Assistants）等に適用することも可能である。また、図1に示した対応関係テーブルのうち、アプリケーション4が所望するファンクションとそのファンクションに対応するリソースの情報は、当該アプリケーション4側がリソース取得要求の際にその都度リソースマネージャ1へ送っても良い。また、優先度は、アプリケーションの優先度だけでなく、ファンクション単位の優先度、更にファンクション内の各リソースの優先度も含まれ、リソースマネージャ1は、ファンクションの優先度に応じたリソースの競合判定や、ファンクション内の各リソースの優先度に応じた競合判定を行い、その競合判定結果に応じてリソースの取得完了等の通知を行っても良い。

【0062】

【発明の効果】

本発明においては、リソースの競合判定や使用許可、リソースの使用権利についての管理はリソース管理装置が行い、リソースを使用していた処理手段がそのリソースを開放したときに、そのリソースについて開放通知要求を行っている処理手段が存在する場合には、その処理手段に対してリソースの開放がなされたことを通知することにより、複数の処理手段（例えば複数アプリケーション）間でリソースを効率的に且つ柔軟に割り付け可能とし、矛盾のなくリソースの排他使用を実現可能である。

【図面の簡単な説明】

【図1】

各ファンクションと各リソースの対応関係の表を示す図である。

【図2】

本発明にかかるリソース管理を実施する電子機器の概略構成を示すブロック図である。

【図3】



本実施形態の電子機器の一例としての携帯電話端末の概略構成を示すブロック図である。

【図4】

優先度が $APP1 > APP2$ で開放通知要求無しの場合のリソースマネージャによるリソースの競合管理処理のタイムフローチャートである。

【図5】

優先度が $APP1 > APP2$ で開放通知要求有りの場合のリソースマネージャによるリソースの競合管理処理のタイムフローチャートである。

【図6】

優先度が $APP1 \leq APP2$ で開放通知要求無しの場合のリソースマネージャによるリソースの競合管理処理のタイムフローチャートである。

【図7】

優先度が $APP1 \leq APP2$ で開放通知要求有りの場合のリソースマネージャによるリソースの競合管理処理のタイムフローチャートである。

【図8】

優先度が $APP1 \leq APP2 < APP3$ で開放通知要求有りの場合のリソースマネージャによるリソースの競合管理処理のタイムフローチャートである。

【符号の説明】

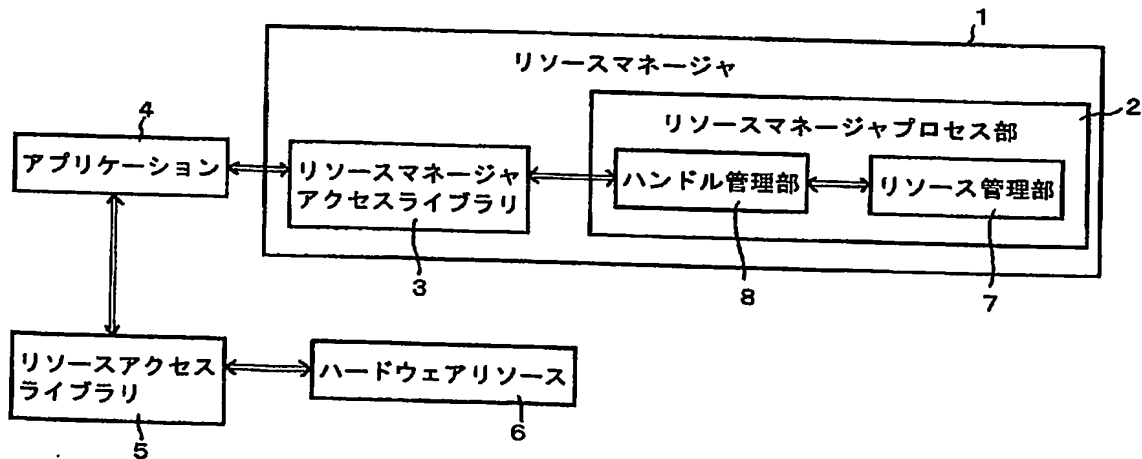
1…リソースマネージャ、2…リソースマネージャプロセス部、3…リソースマネージャアクセスライブラリ、4…アプリケーション、5…リソースアクセスライブラリ、6…ハードウェアリソース

【書類名】 図面

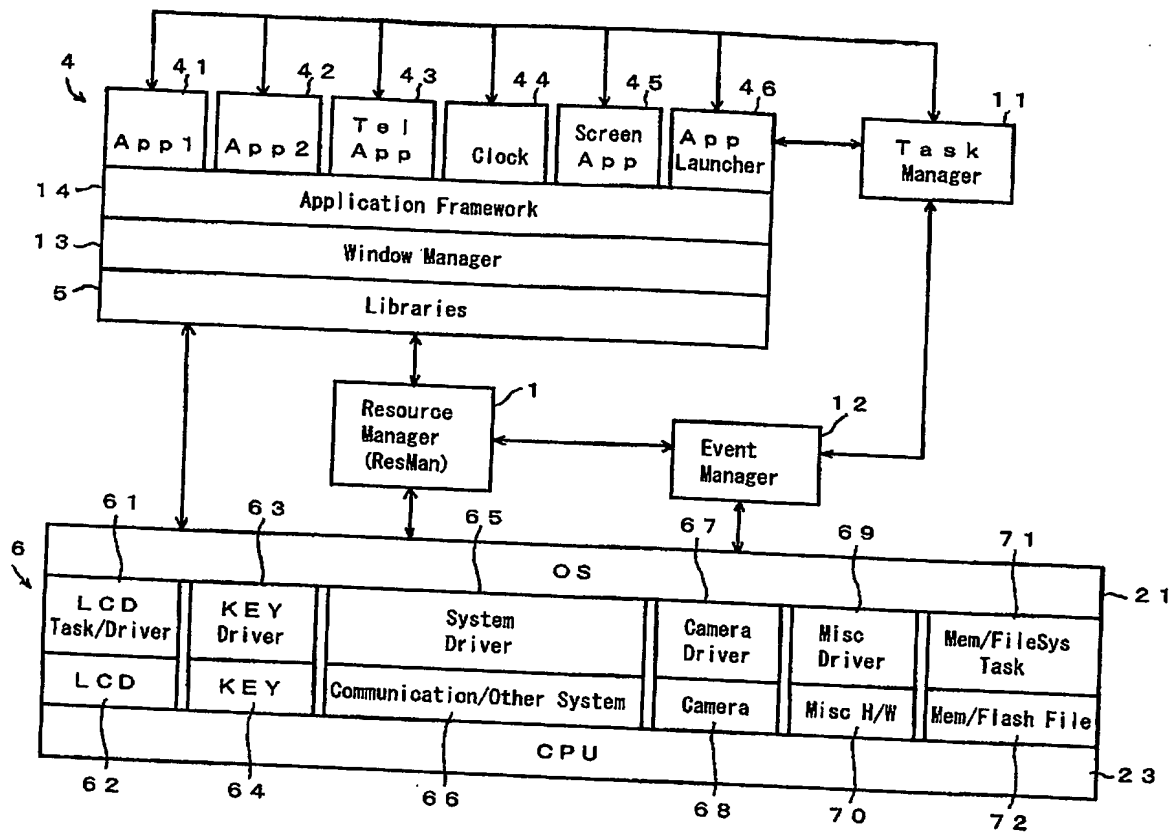
【図 1】

		リソース															
		RA	RB	RC	RD	RE	RF	RG	RH	RI	RJ	RK	RL	RM	RN	RO	
ファンクション	FA			O													
	FB			O													
	FC			O													
	FD			O													
	FE		O														
	FF		O														
	FG								O								
	FH									O							
	FI					O		O		O	O		O				
	FJ	O		O	O	O											
	FK						O	O		O	O	O	O		O		
	FL					O		O		O		O		O		O	
	FM					O		O		O	O	O		O	O		
	FN				O				O	O			O		O		

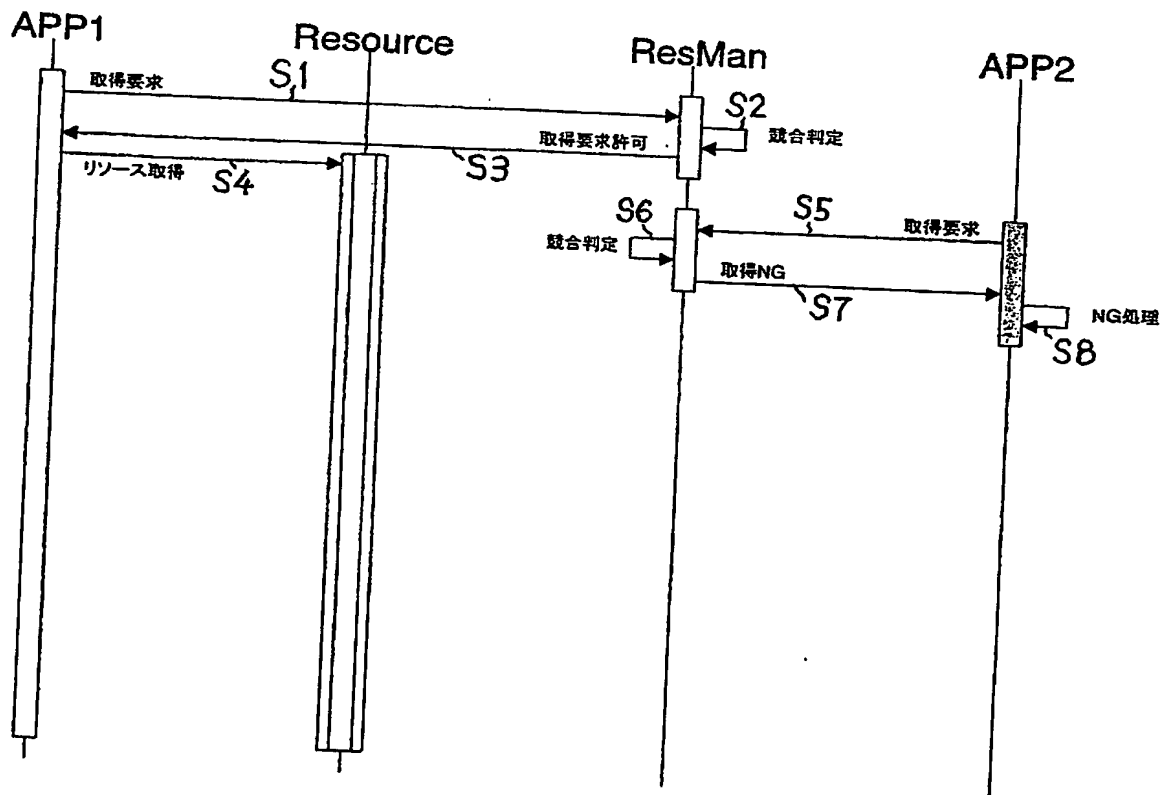
【図 2】



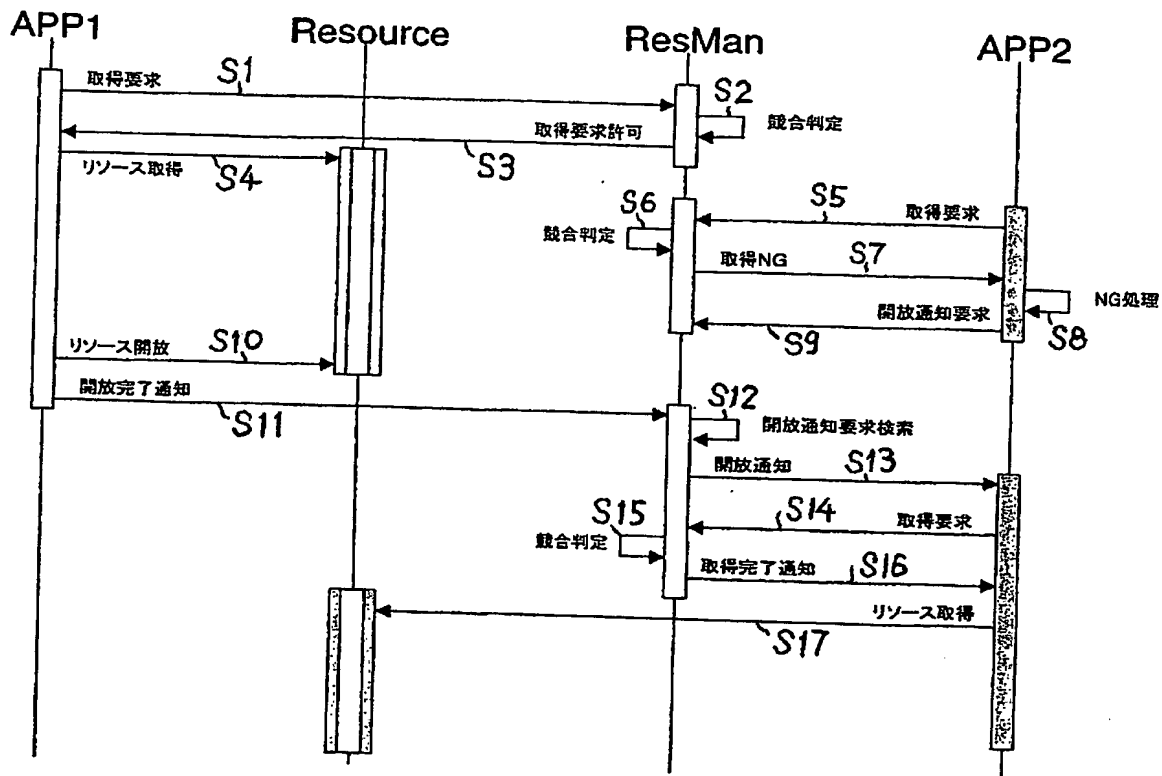
【図3】



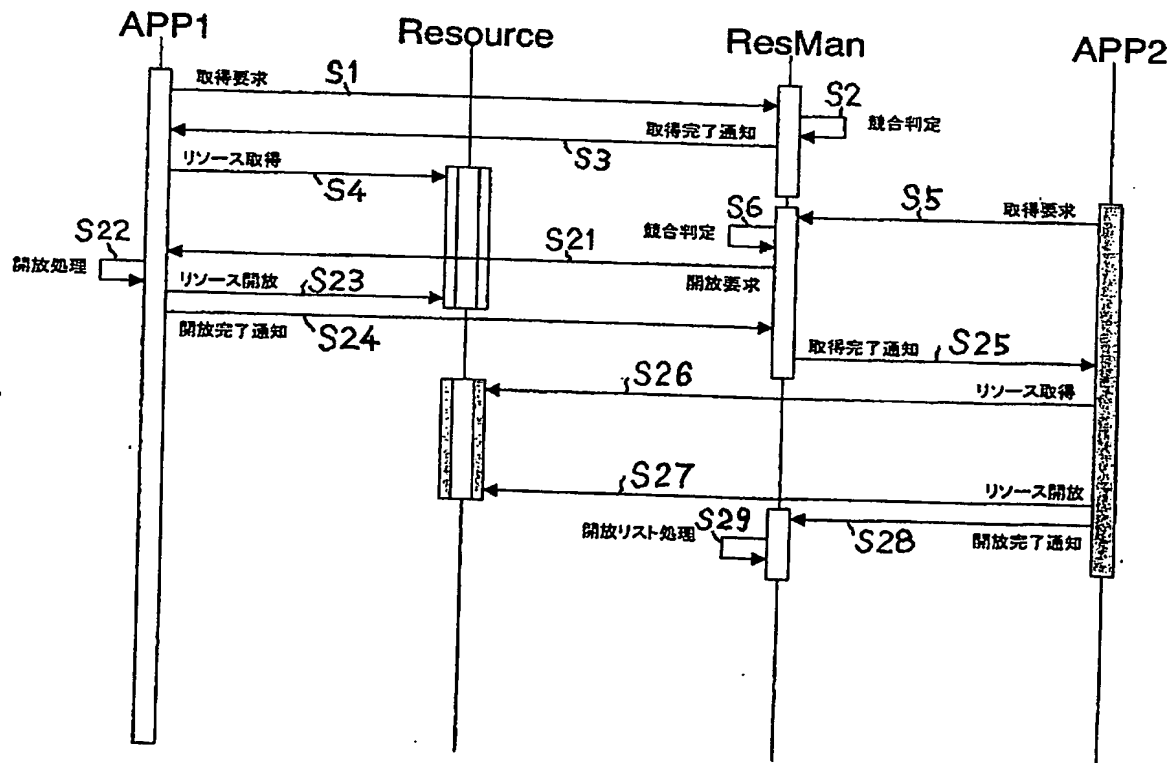
【図4】



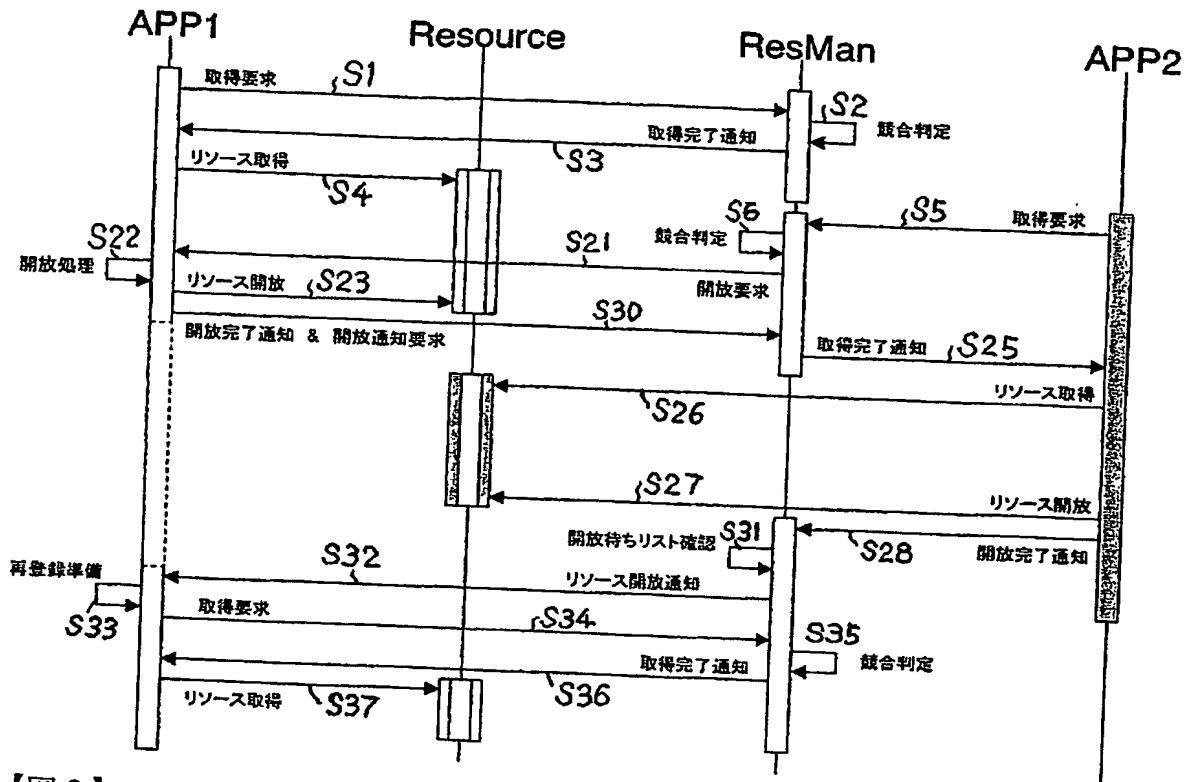
【図 5】



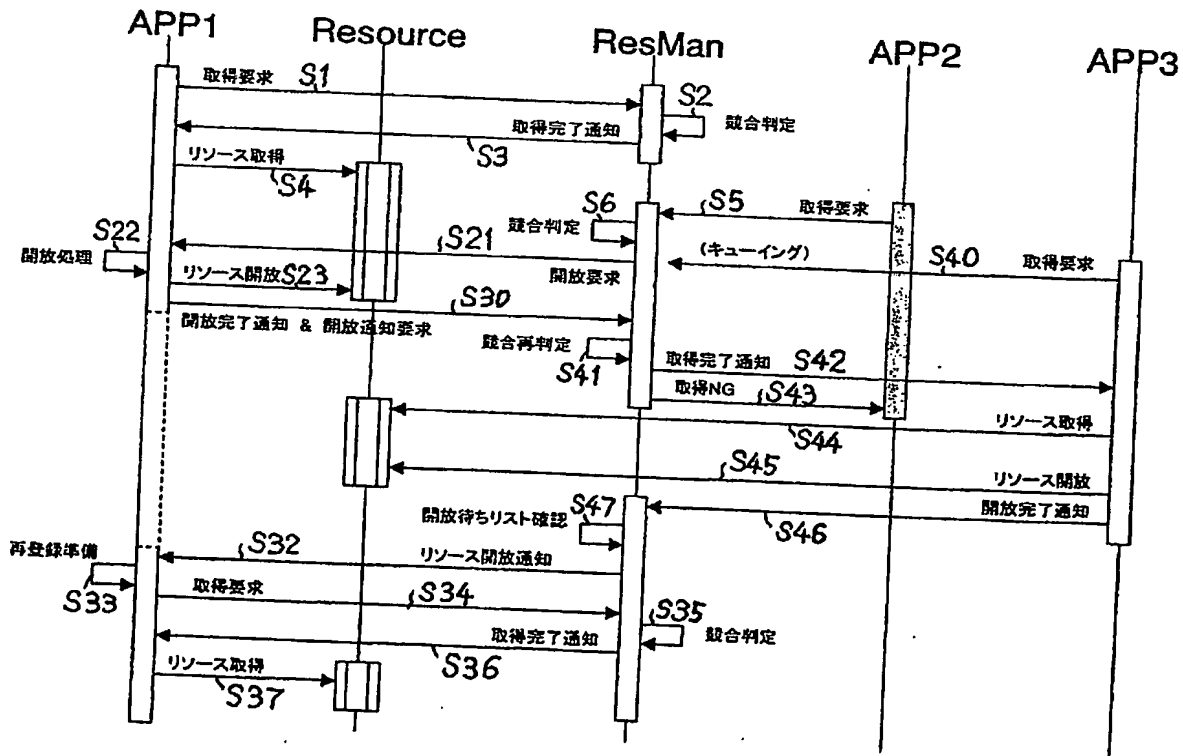
【図 6】

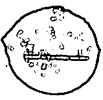


【図7】



【図8】





【書類名】 要約書

【要約】

【課題】 複数アプリケーション間でリソースを効率的に且つ柔軟に割り付け可能とし、矛盾のなくリソースの排他使用を実現する。

【解決手段】 アプリケーションAPP1がリソースを使用しているとき、アプリケーションAPP2からリソースの取得要求を受け取る（ステップS5）と、リソースマネージャ(ResMan)は、アプリケーションAPP1とAPP2のリソースの競合判定を行い（ステップS6）、優先度の低いアプリケーションAPP2には取得NGの通知を返す（ステップS7）。アプリケーションAPP2からリソースの開放通知要求（ステップS9）を受け取った後、アプリケーションAPP1から開放完了通知（ステップS11）を受け取ると、リソースマネージャは、他にリソースの競合（ステップS15）がなければ、アプリケーションAPP2に対してリソースの取得完了通知を発行（ステップS16）する。

【選択図】 図5

認定・付加情報

特許出願の番号	特願 2003-165237
受付番号	50300969986
書類名	特許願
担当官	第七担当上席 0096
作成日	平成15年 6月11日

<認定情報・付加情報>

【提出日】	平成15年 6月10日
-------	-------------

特願 2003-165237

ページ： 1/E

出 願 人 履 歴 情 報

識別番号

[501431073]

1. 変更年月日

2001年11月 6日

[変更理由]

新規登録

住 所

東京都港区港南1丁目8番15号 Wビル

氏 名

ソニー・エリクソン・モバイルコミュニケーションズ株式会社